Programming Languages

2nd edition

Tucker and Noonan

Chapter 3
Lexical and Syntactic Analysis

Syntactic sugar causes cancer of the semicolon.
A. Perlis

Contents

3.1 Chomsky Hierarchy

Regular grammar -- least powerful

Context-free grammar (BNF)

Context-sensitive grammar

Unrestricted grammar

Regular Grammar

Simplest; least powerful

Equivalent to:

- *Regular expression*
- *Finite-state automaton*

Right regular grammar: $\omega \in T^*$, $B \in N$

$$A \to \omega B$$

$$A \to \omega$$

Example

$$\mathit{Integer} \rightarrow 0\ \mathit{Integer} \mid 1\ \mathit{Integer} \mid \ldots \mid 9\ \mathit{Integer} \mid$$

$$0 \mid 1 \mid \ldots \mid 9$$

Regular Grammars

Left regular grammar: equivalent

Used in construction of tokenizers

Less powerful than context-free grammars

Not a regular language

$\{ a^n b^n \mid n \geq 1 \}$

i.e., cannot balance: ( ), { }, begin end

Context-free Grammars

BNF a stylized form of CFG

Equivalent to a pushdown automaton

For a wide class of unambiguous CFGs, there are table-driven, linear time parsers

Context-Sensitive Grammars

Production:

$\alpha \rightarrow \beta \qquad |\alpha| \leq |\beta|$

$\alpha, \beta \in (N \cup T)^*$

ie, lefthand side can be composed of strings of terminals and nonterminals

Undecidable Properties of CSGs

Given a string ω and grammar G: $\omega \in L(G)$

$L(G)$ is non-empty

Defn: *Undecidable* means that you cannot write a computer program that is guaranteed to halt to decide the question for all $\omega \in L(G)$.

Unrestricted Grammar

Equivalent to:

- *Turing machine*
- *von Neumann machine*
- *C++, Java*

That is, can compute any computable function.